

CSV2BUFR Training

Total Learning outcomes

By the end of the session you should be able to use the tools provided by csv2bufr to:

- Create a BUFR mapping file and understand the basic structure
- Understand the links between the mapping file and the data in their CSV files.
- Customise the BUFR mapping file (and/or input CSV) to apply to their CSV data.
- Successfully convert their data to the recommended BUFR sequence to meet GBON requirements.

Essentials

Before attempting the questions below, here are some essential commands that will be helpful:

Mappings Create

The `mappings create` function will allow you to create an empty BUFR mapping template JSON file, which maps the CSV column headers to their corresponding ecCodes element. It can be used in the following way in the command line:

```
csv2bufr mappings create <BUFR descriptors> --output <output_dir>
```

For more information, see an example [here](#).

Data Transform

The `data transform` function is what will convert a CSV file to BUFR, and can be used in the following way in the command line:

```
csv2bufr data transform --bufr-template <my_template.json> --output-dir  
↪ <./my_folder> <my_data.csv>
```

BUFR Dump

The `bufr_dump` function will allow you to inspect the BUFR files created from the conversion. It has many options, but the following will be the most applicable to the exercises:

```
bufr_dump -p <my_bufr.bufr4>
```

This will enumerate the content of your BUFR on screen. If you are interested in the values taken by a variable in particular, you can use the `grep` command. For example:

```
bufr_dump -p <my_bufr.bufr4> |grep -i 'temperature'
```

This will enumerate the variables related to temperature in your BUFR file. If you want to do this for multiple types of variables, you can use the `\|` command. For example:

```
bufr_dump -p <my_bufr.bufr4> |grep -i 'temperature\|wind'
```

Exercises

To begin with the exercises, login to your VM, and start the `csv2bufr` container with the following command:

```
docker run -it -v csv2bufr-exercises:/exercises csv2bufr
```

Exercise 1

Learning Outcomes

- Understand the basic usage of `csv2bufr`:
 - Generate simple BUFR mapping file.
 - Run `csv2bufr` on a test data file and convert to BUFR.
- Ability to lookup up the BUFR element number for a parameter to be encoded.
- Create a basic BUFR mapping file.
- Encode BUFR data for the example input data and inspect the data.

Questions

1. Enter folder `ex_1` using command `cd ./ex_1`. Using the ECMWF eccodes website, create a mapping file of the following variables:

- WIGOS station identifier (series, issuer, issue number, local identifier)
 - Date (year, month, day)
 - Time (hour, minute)
 - Location (latitude, longitude)
 - Barometer height above sea level
 - Station level pressure
 - Air temperature
 - Relative humidity
2. Open the mapping file you have just created, and remove any unused elements. In particular, look for and delete the elements which have "value": "" (you'll notice these elements do not directly correspond to any columns in the CSV file).
 3. Use the `csv2bufr data transform` function to convert `ex_1.csv` to BUFR.
 4. Use BUFR Dump to find the latitude and longitude value stored in the output BUFR file. Verify these values using the CSV file.

Exercise 2

Learning Outcomes

- Understand the relationship between BUFR sequences and elements.
- Create a new mapping file for the data from the previous exercise, but replacing the date, time, and location elements with the appropriate groups.

Question

Enter folder `ex_2` using command `cd .. ; cd ./ex_2`. Repeat the previous exercise, but in part (1) replace the following elements with their respective BUFR sequences (which have form 3XXYYY):

- WIGOS station identifier (series, issuer, issue number, local identifier)
- Date (year, month, day)
- Time (hour, minute)
- Location (latitude, longitude)

Note: the file `BUFR_TableD_en.csv` from the `./BUFR_tables` directory contains the defined BUFR sequences. The first few columns of this file are:

- BUFR category (numeric)
- BUFR category (name)

- **BUFR sequence number**
- **BUFR sequence name**
- BUFR sequence subtitle
- **Included BUFR element number**
- **Included BUFR element name**

The important columns are highlighted in **bold**. A given sequence will appear multiple times, once for each BUFR element it contains.

(**Hint:** Search for the corresponding 6 digit codes you found in the previous exercise, and find the corresponding sequence in the 3rd column. For example, the BUFR elements 005002 (Latitude) and 006002 (Longitude) can be replaced with sequence 301023 (Latitude/longitude (coarse accuracy).)

Exercise 3

Learning Outcomes

- Understand how to customize a mapping template to a simple CSV data file with different column names.

Questions

1. Enter folder `ex_3` using command `cd . ; cd ./ex_3`. Open `ex_3.csv`, and compare this to `ex_2.csv`. You should notice that the data is the same, but the column names are different. With this in mind, create a mapping template file for this data.
2. Open the mapping file, delete the redundant elements, and change the "value" item to correctly correspond to each column in `ex_3.csv`.
(**Hint:** For the first column, "eccodes_key": "#1#wigosIdentifierSeries" should now be paired with "value": "data:wigos_identifier_series".)
3. Use the `csv2bufr data transform` function to convert `ex_3.csv` to BUFR.
4. Check that the data stored in the output BUFR is the same as that in the CSV.

Exercise 4

Learning Outcomes

- Know the sequence for hourly climate data.
- Understand how to create a mappings file for hourly data according to GBON requirements.

Questions

In this question you are free to either work with your own synoptic CSV data, or use the `ex_4_hourly.csv` data found in the `ex_4` folder.

1. Noting that a SYNOP report cannot contain the WIGOS station identifier, creating a mappings template which contains the mappings for both the WIGOS station identifier and the hourly synoptic data.
2. Delete all elements in the mapping template that are not present in the CSV data (you should expect to delete most of the file!), and edit element names appropriately.
NOTE: In this example you will need to make sure `number_header_rows` and `column_names_row` is correct.
3. Convert this data to BUFR.

NOTE: The following are additional questions, time permitting.

Exercise 5

Learning Outcomes

- Know that the “BUFR Manual on Codes” has requirements on the units of variables, e.g. Celsius -> Kelvin.
- Understand how to use the mapping file to convert the units without modifying the original CSV data.

Questions

1. Enter folder `ex_5` using command `cd .. ; cd ./ex_5`. Open `ex_5.csv`. Notice that this file contains the same data and column names as `ex_2.csv`, but uses different units. In particular:
 - `heightOfBarometerAboveMeanSeaLevel` is given in cm.
 - `nonCoordinatePressure` is given in hPa.
 - `airTemperature` is given in °C.

Using the ECMWF eccodes website, find the correct units for these three variables.

2. Find the scale (x , for a multiplication by 10^x) and offset (addition of a constant c) required for each of these conversions.
(**Hint:** As $1\text{cm} = 1\text{m} \times 10^{-2}$, the first conversion requires a scale of $x = -2$ and an offset of $c = 0$.)
3. Open the mapping file `mapping_5.json`. This is the same mapping file as you generated and modified in *exercise 3*. Using your answers in the previous question, convert the units using the `scale` and `offset` keys.
4. Convert `ex_5.csv` to BUFR.

Exercise 6

Learning Outcomes

- Understand how simple quality control measures can be implemented by specifying valid minimum and maximum values in the mapping file.

Questions

1. Enter folder `ex_6` using command `cd ..; cd ./ex_6`. Open `ex_6.csv`. Notice that this time the `relative_humidity` column contains an incorrect value. Justify why it is incorrect.
2. Think of the valid minimum and maximum values for relative humidity.
3. By adjusting the `valid_min` and `valid_max` keys of the `relativeHumidity` element in the mapping file `mapping_6.json`, enforce a quality control measure which prevents this value from being written to BUFR.
4. Add more valid minimum and maximum values to the mappings file according to your own preference.
IMPORTANT: The minimum and maximum values must have the **same units** as your **original** input data, not the converted values discussed in *exercise 5*.
5. Convert `ex_6.csv` to BUFR, and use BUFR Dump to verify that this variable has no value in the output BUFR.